CDF note xxxx
October 08, 2004

CDF production farm: performance and upgrade

Y.C. Chen[a], S. Hou[a], T.L. Hsieh[a], R. Lysak[a], I.V. Mandrichenko[b],
J. Syu[b], S.C. Timm[b], S.A. Wolbers[b],

[a] Academia Sinica, Taipei , Taiwan
[b] Fermi National Accelerator Laboratory, Batavia, Illinois

Abstract

The production performance is reported for reprocessing with cdfsoft 5.3.1 and data newly taken. With the experience running the farm in full capacity, we project the computing requirement for upgrade and migration to the SAM metadata platform.

http://fnpcc.fnal.gov/SamFarm/farm2004.pdf

1 Introduction

The CDF production farm provides data processing with a cluster of about 200 dual Pentium nodes for a total of 770 GHz CPU power. The current CDF data logging rate is around 10 pb^{-1} per week with less than 10 stores of $p\bar{p}$ collisions corresponding to a few tens of physics runs. The farm has a priority of catching up data processing provided the calibration is ready. It has free CPU cycles which are put into use for data reprocessing.

In this note we first describe the farm architecture, the experience of reprocessing with cdfsoft 5.3.1 for the "0d data-set, and the capacity moving data tape-to-tape. The farm has a successful operation processing about 2 TB data per day. Problems encountered are mostly book-keeping related for cleaning corrupted files of various reasons. For upgrade, we discuss the scaling issue related to file-tracking, disk space, and data through-put rate. We propose for developing management tools for running data production on a SAM metadata platform.

2 Farm architecture

The CDF production farm has two server nodes cdffarm1 and cdffarm2 that host control daemons and an MySQL data base server. The operation interface includes a java server running on cdffarm2 and a web server node fnpcc. The cdffarm1, a six years old SGI machine soon to be replaced by a Dell server, is the core running the FBS batch system and the MySQL data base. The cdffarm2 has control daemons for resource management and job control for each data stream. The disk space is a "dfarm" file system, which is a collection of IDE hard-disks of all worker nodes. The dfarm management is hosted on cdffarm1. At present the dfarm capacity is 23 TB.

The flow chart of file control is illustrated in Fig. 1. The CDF production farm is a complete chain of data processing tape-to-tape from the Enstore storage. File records are fetched and written to Datafile Catalog (DFC). The farm has a internal data base on MySQL used for task control, tracking of file in process, and the history of them. Jobs are submitted by daemons to a FBS batch system running on worker nodes.

The worker nodes purchased over the years are list in Table 1. Old nodes were replaced after three years service. At present we have 174 workers of a total 680 GHz (P3 equivalent) in service including 64 dual P4 2.6 GHz machines added this spring. There are 16 dedicated IO nodes equipped with optical giga-links. These nodes are configured

Year	Numbers	Type	P3 equivalent GHz
2001	64	P3/1.00 duals	128
2002	32	P3/1.26 duals	81
2002	32	AMD/1.67 duals	107
2003	64	P4/2.6 duals	450*

Table 1: Farm CPU purchased over the years. These are the nodes in use. (* scaled by 1.35 to P3 equivalent).

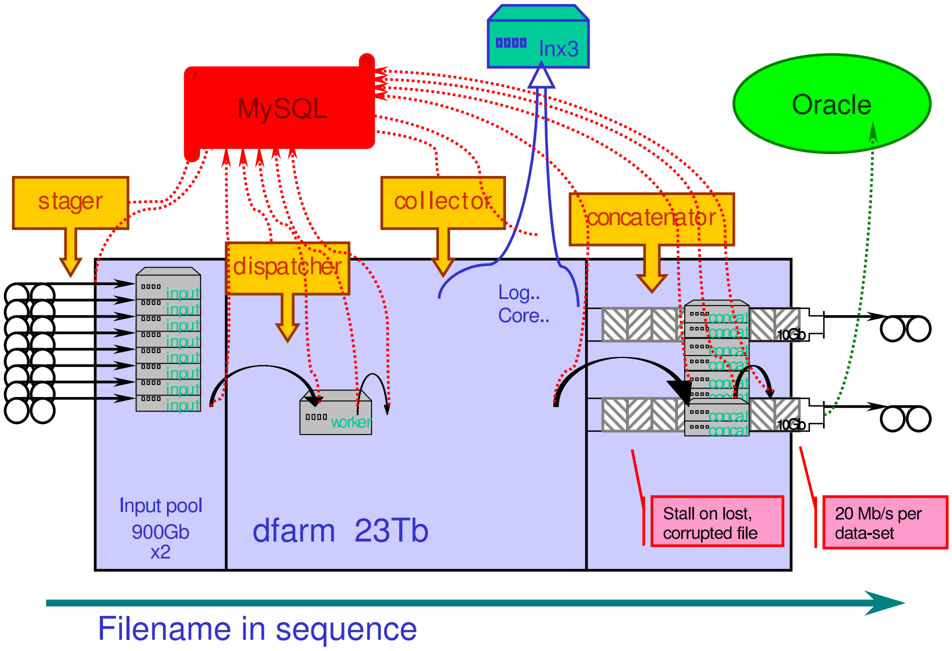

Figure 1: File control in the CDF production farm.

with the pnfs file system for access to the Enstore storage. We have recently added a Cisco switch module to provide twist-pair giga-links for a total 48 ports. Later we may configure more IO nodes for the farm. IO to Enstore is the ultimately limit for the farm performance.

The farm processes are divided in farmlets each consumes a specified "data-set". Jobs are dispatched in units of "file-set". A file-set is a pnfs subdirectory of 10 files, typical file size is 1 GB. Raw data of Stream A to J are processed independently in farmlets. Eight of the IO nodes are input stagers copying data from Enstore tapes to their scratch disk area, and then into the dfarm. The staged raw data files are first dispatched to workers running CDF ProductionExe and the outputs are copied to dfarm. A raw data file can have multiple outputs of different physics data-sets, and the file size varies from 20 MB to 1 GB. The eight output concatenators collect products of each data-set in dfarm, execute AC++dump for concatenation into 1 GB output files to be copied to Enstore tapes.

Each farmlet has a series of daemons tracking the file status from stage-in to storage and making records in the internal data base. The history of them are recorded also. The daemons check the history to decide on job submission. For example, a consumed file won't be staged again, and a file in process will be waited for concatenation in sequence. The data base record is not and probably can never be instrumented to realize an abnormal failure for a job or a missing file. This is a major issue considered for upgrade. We have also observed heavy load fetching MySQL records on the old cdffarm1 server. It requires attention for hardware upgrade also.

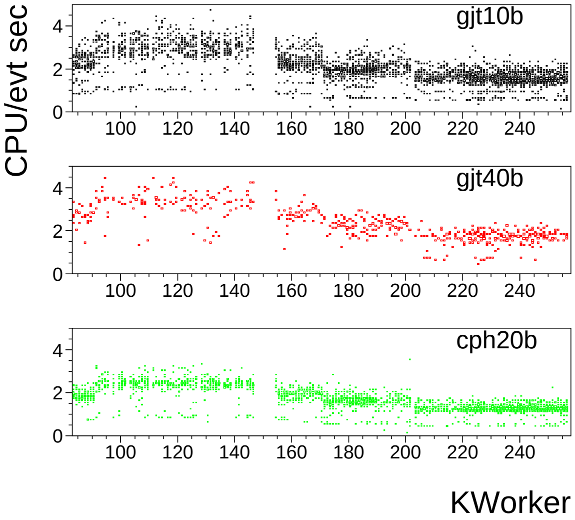

Figure 2: CPU monitored for jet (gjt0b) and photon (cph0b) events.

3 Farm capacity

Capacity of the CDF production farm may be presented for experience of data reprocessing with cdfsoft version 5.3.1. The task was launched in early March 2004 for processing "0c" primary data-sets into version "0d". Tests were made to evaluate the farm capacity for processing 400 M events in a month. The event size and CPU time varies for data-sets. In Fig. 2 the CPU time per event is plotted for all worker nodes for jet and photon data. On average the computing time on a P3 1 GHz machine varies from 4.5 sec for jet events to 2.5 second for photon events.

The speed staging data from tape depends on how fast the link to Enstore movers is established. Once a mover is allocated, staging a file-set of 10 GB takes about 20 minutes. Staging output is more a concern for the limited dfarm space and network speed collecting files to concatenation node. In addition, the concatenation makes intensive data base access recording the file status. The tape access is limited for having one mover writing a data-set at a time. Having all workers producing files of the same data-set would certainly out pace the data logging rate. Therefore it is necessary to optimize stage-out rate by increasing the number of concatenation jobs and the number of data-sets. The instant tape writing rate is 30 MB/sec, however, average over the latency establishing network link, the data transmission rate drops to below 20 MB/sec.

Study of the data logging rate is demonstrated in Fig. 3 for a test of logging three outputs of primary data-sets and the solo output of Stream-I farmlet. Concatenation is not required for reprocessing of primary data-sets, therefore the output logging rate represents the tape writing speed and latency. In this test we could submit up to eight concatenation jobs waiting in queue for one Enstore mover on one data-set. The test began with two data-sets each having two jobs running, that is, two movers were allocated each allocated by two output nodes. The integrated data moving speed observed is 18 MB/sec. By adding one more job for each data-set, the mover was effectively kept in writing, and the data transmission rate for a data-set increases to 15 MB/sec. If concatenation is

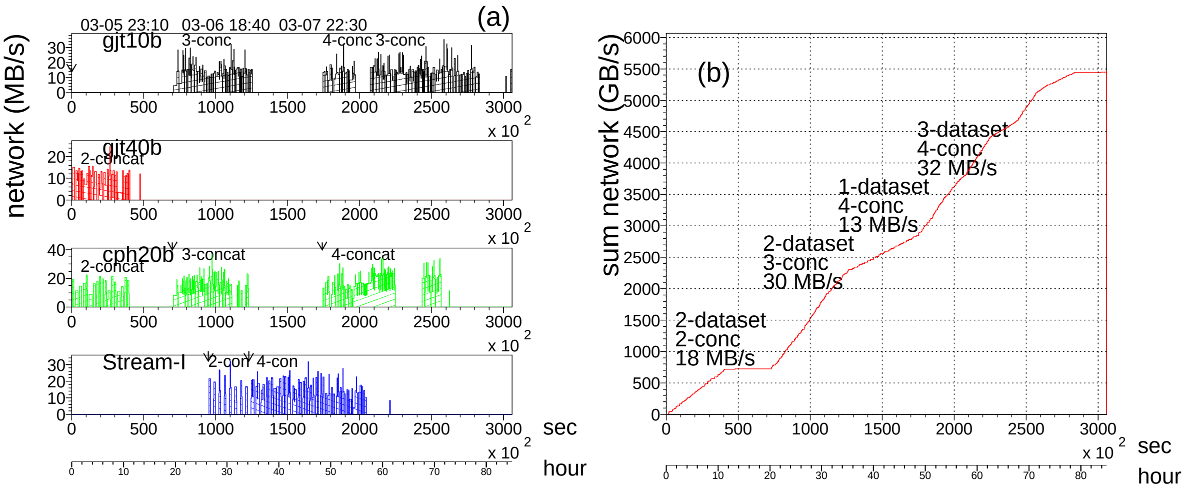

Figure 3: Data logging rate for stage-out of four data-sets running two to four concatenation jobs. The instant rates of each data-set are shown in (a) and the integrated in (b).

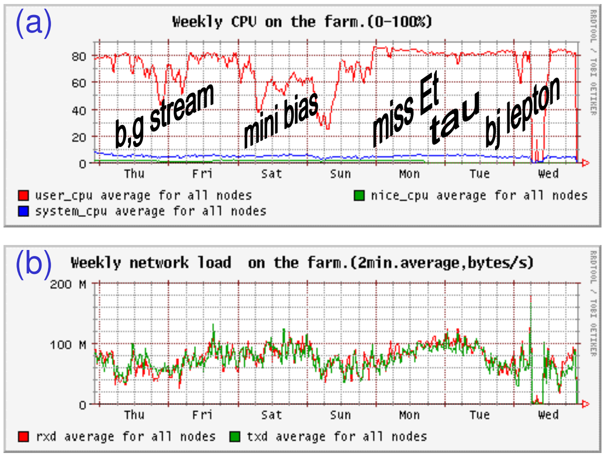

Figure 4: (a) CPU load and (b) dfarm traffic of the week of March 18-25, 2004.

required, the additional CPU cycle takes about 20 to 30 minutes and the job stays twice longer on a concatenator. The Stream-I tested was running four concatenations to reach a data transmission rate of 13 MB/sec.

The 5.3.1 reprocessing was performed on five farmlets. Once a farmlet process was submit, the stage-in for all the 0c files of the data-set was submitted in an instance. As a consequence, the data reprocessing was carried out very much in queues of data-sets. The CPU usage at a glance for the week of March 18 is shown in Fig. 4. A lag in stage-in was observed when the farm process was switching to a new data-set. It is seen as the dips in CPU in Fig. 4.a, fot the lack of input files. File-sets are distributed close in sequence in a tape. The lag at the beginning of loading a data-set is because files requested are stored in the same tape, therefore all stage-in jobs are waiting in queue for one tape. Overall the stage-in was effective feeding data to dfarm. The CPU usage varies for data-sets. The “minimum bias” data-set is a example showing smaller file size and CPU time in

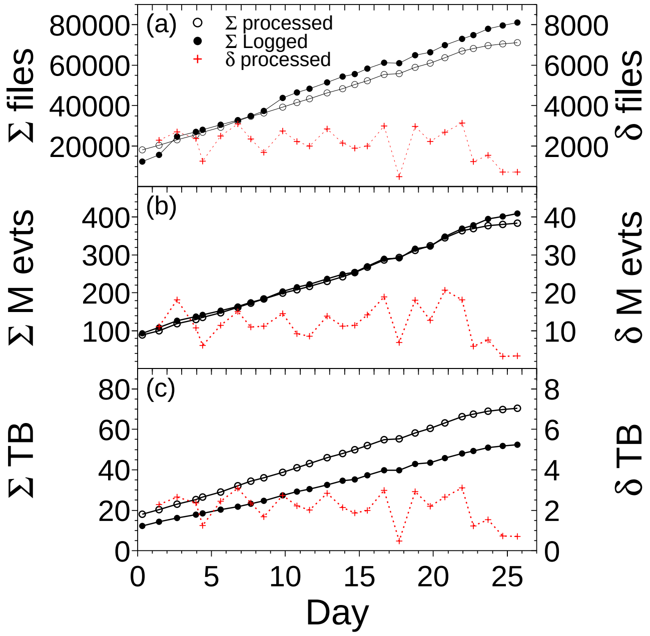

Figure 5: Reprocessing statistics are shown for (a) number of files, (b) number of events, and (c) storage logging rate. Records were made daily. Plots are shown for the integrated and the day-to-day rates.

comparison with the jet data. The CPU per event is about 40% less, and the stage-in rate was not able to catch up the CPU consumption.

The last "0c" data-set in reprocessing was the "hbhd" finished on May 10. The data logging rates of the 5.3.1 reprocessing are shown in Fig. 5 for the number of files, number of events, and total file size written to Enstore. The event size of "0d" output was trimmed by about 30 %. On average we were putting over 2 TB per day for over 10 M events into the Enstore storage.

The "0d" production continued for post-November '03 data of Stream-H. It began in late May and was finished for final logging in mid-June.

4 Farm catching up newly taken data

The farm processes are constantly checking the on-line data base for newly taken data to be processed. The timely processing is critical for detector monitoring. The Stream-F is operated for BeamExe beam line calibration used for pass-11 calibration. Outputs of Stream-A are used for data-quality check. And data-sets of Stream-B and G are used for higher level calibration. Data processed with pass-13 or higher calibration are qualified for physics analysis.

The load in the farm for newly taken data, for the recent logging rate of 10 pb^{-1} per week, is less than half used. Shown in Fig. 6 is the CPU load of the week of July 1, 2004, where the first two days were fully loaded for backlog followed by small fractions of occasional usage. The raw-data volume collected by CDF is shown in Fig. 7 in red. The processed data volume is shown in blue. In February the COT had unstable gain.

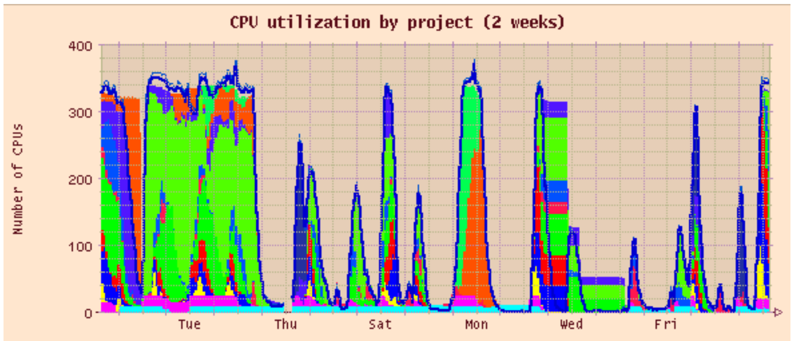

Figure 6: CPU load after 5.3.1 reprocessing is (Thu July 1, 2004) relatively light.

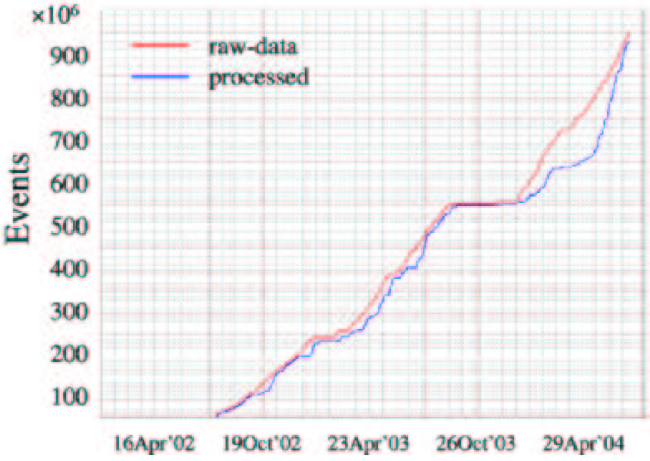

Figure 7: The cumulative raw-data and production data rate.

Raw data processing was held except for detector study. Meanwhile the farm was put in use for 5.3.1 reprocessing. Raw-data processing resumed in early May for post-November Stream-H data to complete the `0d` data-set of 5.3.1 production.

Data collected after February 2004 have being processed with pass-11 calibration, and stored in the "`0e`" data-set. Data of this version will need to be reprocessed again for high level calibration.

5 Farm upgrade

In the 5.3.1 reprocessing, we have gain experience for what may fail and we have explored the limit of the farm architecture. Constraints and problems observed are:

- The file tracking imposes multiple threads on MySQL for each input file till it is consumed. There are even more threads being put on a product file through the concatenation till it is written to Enstore. Since hardware and software failure do occur, and there is no presumption for backward cleanup. Therefore a sudden failure for a file in process leaves corrupted records blocking further process.

- MySQL server on the SGI server shows a load at a alarming rate while having multiple concatenation jobs running in parallel that requires record changing for up to a few thousands files. This is presumably resolved after moving the MySQL over to the new Dell server. The load shown in Fig. 8 has dropped to below two.
- The dfarm as a collection of hard-disks on workers echos a search for a file request. It does spin all hard-disks to establish a transaction, therefore it has an essential limit for scaling. The present farm capacity is 23 TB including Raids on three servers. Any hard-disk failure is propagated and observed for a slow dfarm response. We suggest to constrain the dfarm within 20 TB. Raid is not favored either. In a few occasion we had IDE failure in a Raid and resulted to a file transmission rate down to only a few MB/sec while waiting for the Raid being rebuilt.
- Concatenation requires data files in sequence and the outputs are tailored into 1 GB files in size. A problem occurs frequently is that a job failed and the products are being waited for. As a consequence the concatenations stall. If a concatenated file is lost, the problem is even worse. As it has an arbitrary number of input records with the latest truncated to fit for the 1 GB requirement, it becomes very difficult for recovery.
- Demand for human intervention is frequent and time consuming. The system control depends on many daemons and all being running healthy. Some of the daemon failure were fixed by automatic recovery. Most of them require expert check up still. The fix for corrupted file records in MySQL requires detailed understanding of file control and correct diagnosis of the problem. Since the MySQL records are the core of book keeping, upgrade for replacing MySQL shall be demonstrated for advantage.

The experience gained in the 5.3.1 reprocessing demonstrated the heavy load on file-tracking for up to 10,000 files a day. The farm makes elaborated file tracking through the long chained process tape-to-tape. The file tracking has no tolerance for error. The frequent demand for human intervention could be the most expensive part in operation for the farm. The dfarm having a total 23 TB is capable of buffering data for three days

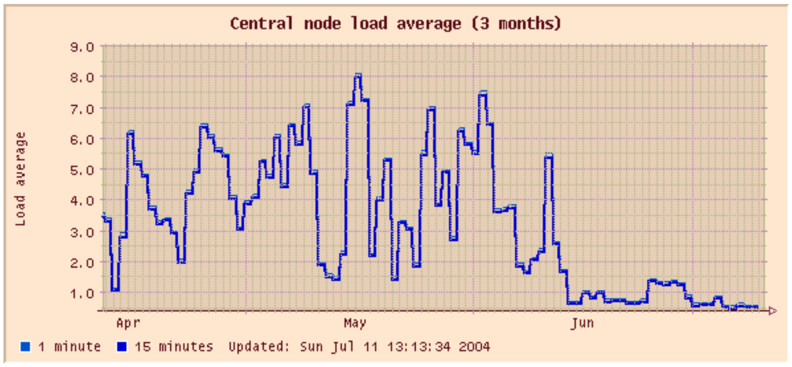

Figure 8: The SGI cdffarm1 node was running the MySQL server till the beginning of June. The load dropped significant after the MySQL was moved to a new Dell server.

(two copies per file are stored in dfarm, thus a factor four storage required for producing one output). Given the limited number of Enstore movers and the restriction of having one mover per data-set, the output rate of the farm can not be scaled easily.

In summary, the farm is in a status saturated for file-tracking and data transmission of about 2 TB a day. The CPU usage was tailored specific for the farm and the data flow was designed rigid to suffice for Enstore IO and file size in concatenation. These demands are loosening, future upgrade can gain advantage making a simpler architecture.

6 Proposal for a SAM farm

Hardware technology and software tools are evolving fast since the farm was developed. For the farm upgrade, we look into a configuration common in CDF. The CDF computing has a fast grow on CAF system in the past two years. The file records are migrating to the SAM metadata system.

The SAM farm in concept is a simplified and modular system compatible to all CDF platforms. We may take the advantage of the SAM metadata and tailor an application package running production on a CAF platform. The benefits are listed in the following:

- SAM is a common platform for CDF, all the CAFs are SAM-stations.
- A SAM project running production on a data-set fetches files from metadata. Therefore the job control is migrating from counting files to counting data-sets.
- A SAM project is a planned submission knowing inputs, and the jobs are submitted in batch queues. Active daemon management is not required.

The present farm manages files using the MySQL data base for the file status on the dfarm space. Queries are made to DFC while copying file-sets to Enstore. All these in combined are what the SAM metadata can offer. The task management will have to be changed from active daemons running the MySQL to batch mode. The book keeping will become a SAM history archive. A private user can run these tasks manually. For the farm processing thousands of files a day, we need a management package fetching SAM project information compatible to what the present farm web page is doing.

The SAM farm manager is a dispatcher setting up job plans for SAM projects. It may have a web-like control interface for job submission, and a web browser for project status and history. In comparison with the present farm, the changes to be made are

- Drop daemons that constantly checking on new entries on file-tracking
  → use watch dogs or cron jobs checking on progress of SAM projects and make further actions.
- Drop internal data base
  → use SAM metadata as the only data base
- Drop java farmlet configuration
  → use options for a SAM projection submission.

7 Summary

At present the production farm capacity is twice of more of the data taking rate of 10 pb^{-1} per week. The hardware upgrade has been following the CDF computing plan for replacing retired nodes. The present farm architecture is IO bounded, and is saturated for 10 M events moving 2 TB data per day. We propose for developing a SAM farm. The SAM farm model allows the data production to be processed on a common platform to the CDF computing facilities.